组会

21.04.24 许晓丹

EACL 2021

Expanding, Retrieving and Infilling: Diversifying Cross-Domain Question Generation with Flexible Templates

Xiaojing Yu
Texas A&M University
vicky_yu@tamu.edu

Anxiao (Andrew) Jiang
Texas A&M University
ajiang@cse.tamu.edu

Motivation

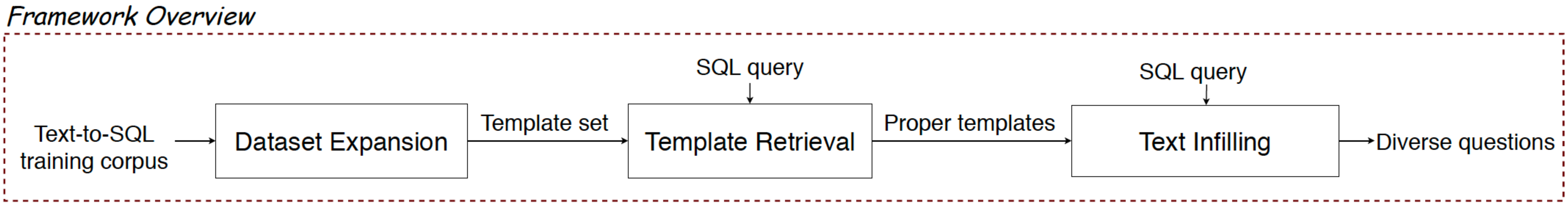
1. 在神经方法中，VAE、beam search等要靠牺牲生成的质量来提高生成的多样性；
2. 在基于规则的方法中，template往往非常严格，且需要预定义大量内容。

↓

把以上二者结合起来：

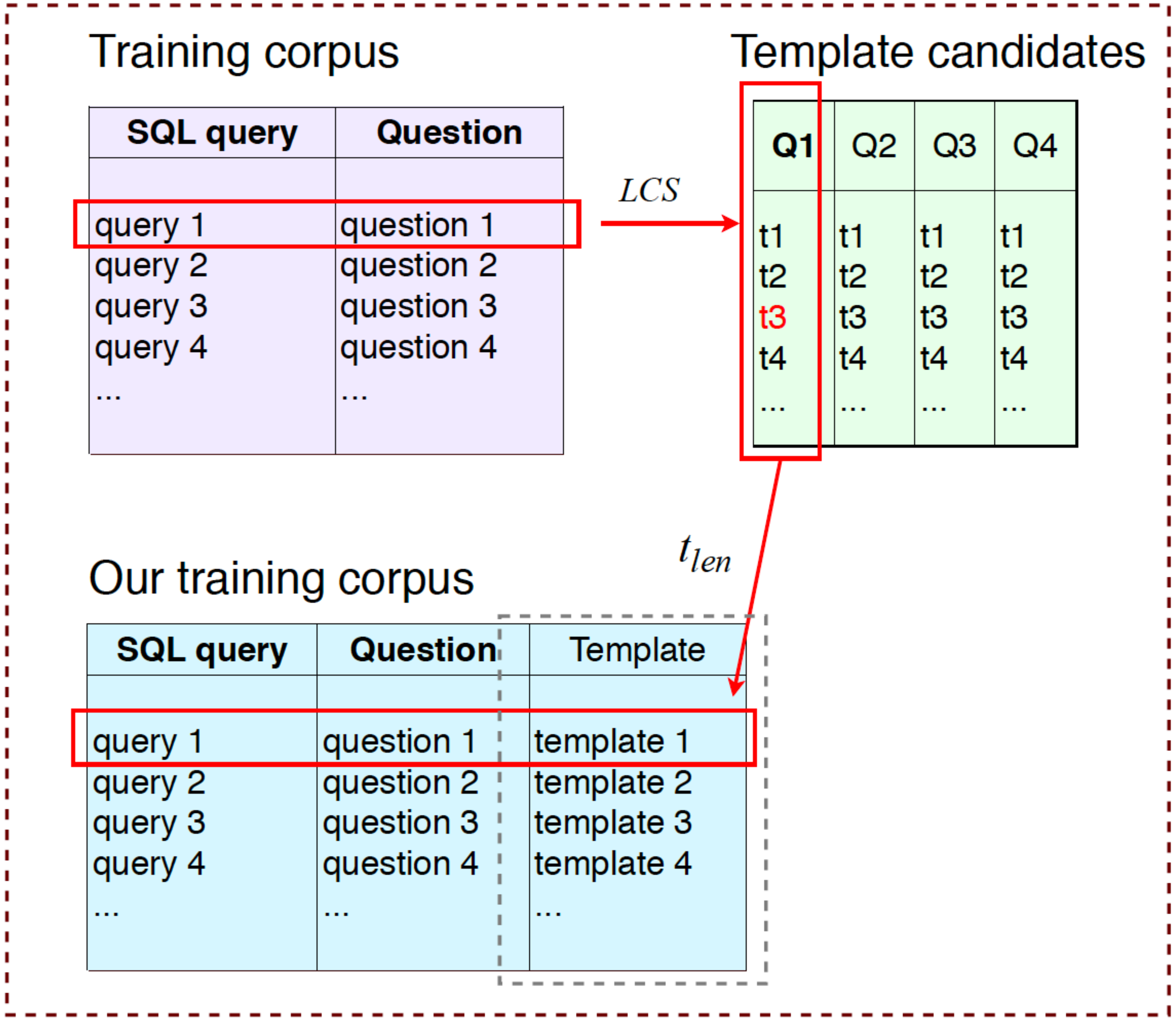
数据集中收集flexible template，再用神经方法去填充内容细节。

Model


Framework Overview
SQL query
SQL query
Text-to-SQL
training corpus
Dataset Expansion
Template set
Template Retrieval
Proper templates
Text Infilling
Diverse questions

Model——Expand

Stage 1: Dataset Expansion



LCS

SELECT COUNT(PLAYER) WHERE (STATE = 'Texas')

SQL query: SQL pattern + content words

Question: template pattern + content words

Model— —Expand

Algorithm 1 LCS-based Template Extraction

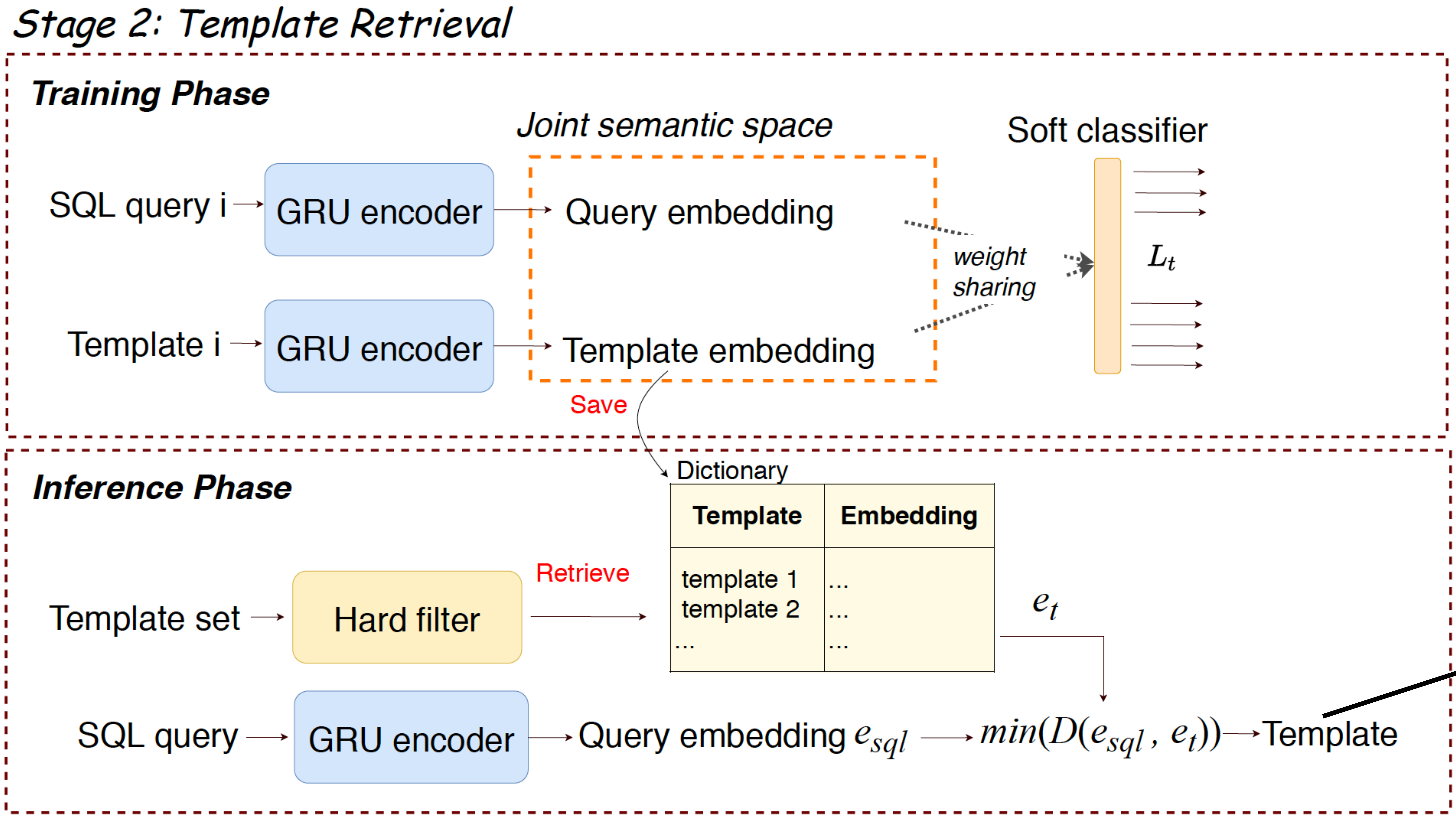
Input: question set $Q = [q_1, q_2..., q_M]$; keyword set W

Output: Template set: T_{len}

for all $q_i \in Q$ **do**
  Initialize dictionary d_i
  for all $q_j \in Q$ **do**
    $c = LCS(q_i, q_j)$
    if $c \bigcap W \neq \emptyset$ **then**
      if $c \notin d_i.keys$ **then**
        $d_i[c] = 0$
      end if
      $d_i[c] += 1$
      Record position index for content
    end if
  end for
  for all $c \in d_i.keys$ **do**
    if $d_i[c] < 20$ **then**
      delete $d_i[c]$ from d_i
    end if
  end for
  $t_{len} = \arg\max_c(length(d_i.keys))$
  Update T_{len} by adding t_{len}
end for
return T_{len}

- Each template should appear over 20 times.
- Each template should includes at least one of the keywords: *where, what, which, when, why, who, how, name, tell* .

Model——Retrieve



训练joint SQL-Template space,

计算两个模态之间的语义距离

选取top-k

Model——Retrieve

Instance-level-Classification

$$P(\cdot|x) = Softmax(W_s^T tanh(e_x))$$
$$P(\cdot|t) = Softmax(W_s^T tanh(e_t))$$

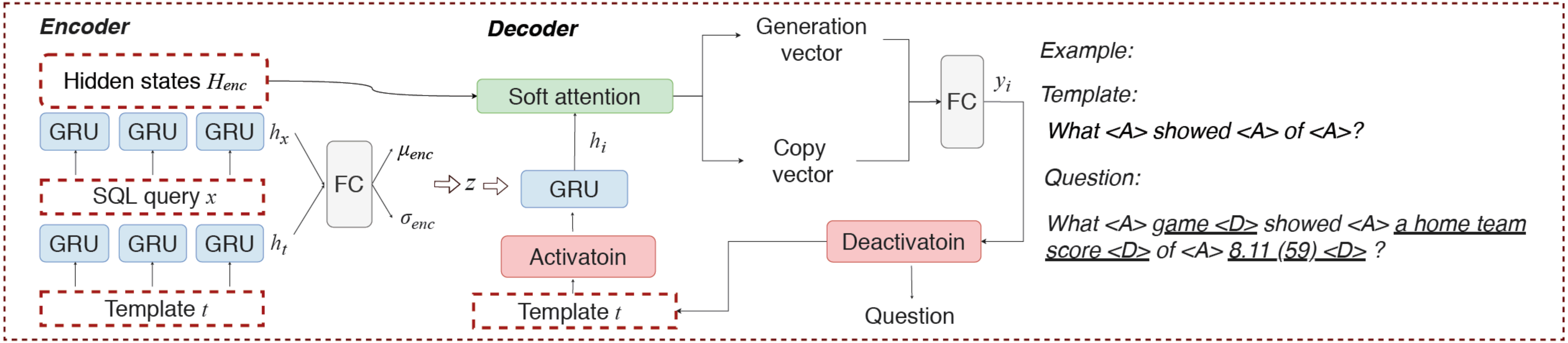
$$\mathcal{L}_t = -\sum_{(x,t,n)\in S} (logP(n|x) + logP(n|t))$$

Model——Generate

generation分解成一系列sub-generation，由template token分隔开；

decoder需要在content-filling state和template-copying state之间切换。

Stage 3: Text Infilling



Model——Generate

用<A>/<D>对question和template进行重写；

Template: `<BEG> Which <A> has the largest <A> ? <END>`
Question: `<BEG> Which <A> one <D> has the largest <A> population among U.S. cities <D> ? <END>`

$$\widehat{y}_i = \begin{cases} Softmax(GRU(\widehat{y}_{i-1}, h_{i-1}^{dec})), & s_i = 1, \\ t'_p, & s_i = 0. \end{cases}$$

用s来维护目前的state；

$$s_i = \begin{cases} 1, & \widehat{y}_{i-1} =< A >, \\ 0, & \widehat{y}_{i-1} =< D >, \\ s_{i-1}; otherwise. \end{cases}$$

Model— —Diverse

为了增加内容表达的多样性，引入隐变量z：

$$\mathcal{L}_q = -E_{z\sim Q}(\sum_{i=1:N} log(P_{\theta_q}(y_i|x,t',z,y_{1:i-1})))$$

$$+ D_{KL}(Q(z|x,t')||P_{\theta_q}(z))$$

多样性：

sentence structure；

content-filling

Experiments

DataSet: WikiSQL / Spider

Models	Quality			Diversity		
	Coverage ↑	ParseAcc ↑	maxBLEU ↑	Self-BLEU ↓	Self-WER ↑	Distinct-4 ↑
QGLV	70.50	**73.89**	37.75	92.86	17.39	33.46
TEMPS	11.34	3.38	5.36	84.50	36.49	59.34
BEAMS	71.49	68.09	**42.17**	79.80	37.39	54.97
ERIT(ours.)	**72.44**	72.79	28.43	**56.30**	**67.00**	**78.73**
w/o lv	70.28	69.53	24.30	57.96	64.39	75.31

Table 1: Automatic evaluation for diverse question generation on WikiSQL. **w/o lv** refers to our model without incorporating the latent variable.

Experiments

Models	Quality			Diversity		
	Coverage ↑	ParseAcc ↑	maxBLEU ↑	Self-BLEU ↓	Self-WER ↑	Distinct-4 ↑
QGLV	13.76	3.97	14.45	96.90	11.73	38.16
TEMPS	6.58	3.87	4.00	**59.25**	7.59	19.73
BEAMS	13.68	4.16	**15.68**	89.39	21.12	50.17
ERI(ours.)	**15.89**	**18.09**	14.42	67.41	**53.23**	**66.96**
w/o lv	12.67	16.70	12.62	62.25	55.58	64.51

Table 2: Automatic evaluation for diverse question generation on Spider. **w/o lv** refers to our model without incorporating the latent variable.

Models	Fluency ↑	Consistency ↑	Diversity
QGLV	**4.56**	**4.64**	1.63
TEMPS	1.13	1.31	1.81
BEAMS	4.16	4.25	2.34
ERI	**4.56**	3.68	**4.31**

Table 3: Human evaluation results.

Experiments — —Ablation

Models	BLEU↑	NIST↑	ROUGH↑	METEOR↑
QGLV	32.19	4.74	64.02	64.25
***Graph2Seq**	38.97	-	-	-
ERI	**48.12**	**5.24**	**76.52**	**75.76**
w/o A/D	43.30	4.85	72.41	73.76
w T	31.42	4.18	63.44	63.94
w/o T	29.76	4.05	62.37	63.17

Table 5: Performance on different sub-module combinations on WikiSQL. *: the value is cited from Xu et al. (2018)

Models	BLEU↑	NIST↑	ROUGH↑	METEOR↑
QGLV	12.60	2.39	44.37	38.74
ERI	**21.30**	**3.11**	**53.83**	**51.04**
w/o A/D	19.02	2.93	52.45	50.41
w T	12.40	2.41	45.8	39.99
w/o T	13.36	2.35	45.05	40.27

Table 6: Performance on different sub-module combinations on Spider.

ICLR 2021

Retrieval-Augmented Generation for Code Summarization via Hybrid GNN

Shangqing Liu[1]*, Yu Chen[2]†, Xiaofei Xie[1]†, Jingkai Siow[1], Yang Liu[1]
[1] **Nanyang Technology University**
[2] **Rensselaer Polytechnic Institute**

Motivation & Work

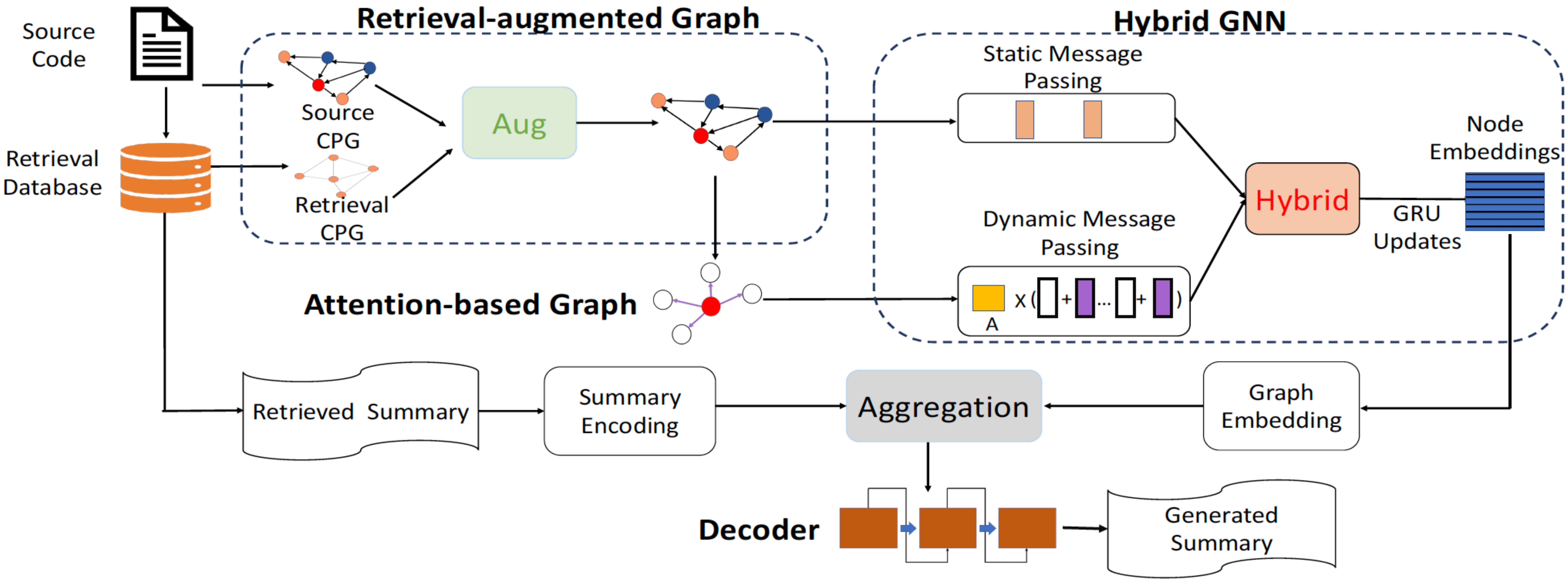
1）Retrieval-based——泛化能力不强；

2）Generation-based——更好的泛化能力，但是无法利用相似的样例；

1. retrieval-augmented mechanism 结合以上两种方法的优点；

2. 为了克服GNN捕获全局图结构的限制，提出了Hybrid GNN，融合attention-based dynamic graph和static graph。

Model



Source Code
Retrieval Database
Retrieval-augmented Graph
Source CPG
Retrieval CPG
Aug
Hybrid GNN
Static Message Passing
Dynamic Message Passing
A
Hybrid
Node Embeddings
GRU Updates
Attention-based Graph
Retrieved Summary
Summary Encoding
Aggregation
Graph Embedding
Decoder
Generated Summary

Model——Retrieval-augmented Static Graph Construction

首先把code转换成Code Property Graph——built on abstract syntax tree (AST)

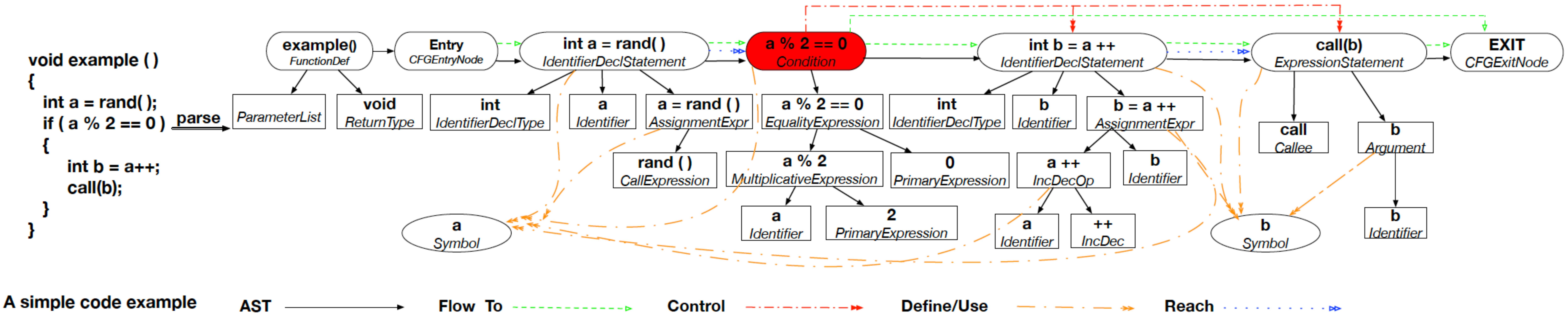


Figure 2: An example of Code Property Graph (CPG).

Model——Retrieval-augmented Static Graph Construction

相似度打分： $z = 1 - \frac{dis(c,c')}{max(|c|,|c'|)}$

计算attention：$\boldsymbol{A}^{aug} \propto \exp(\mathrm{ReLU}(\boldsymbol{H}_c\boldsymbol{W}^C)\mathrm{ReLU}(\boldsymbol{H}_{c'}\boldsymbol{W}^Q)^T)$

$$\boldsymbol{H}_c' = z\boldsymbol{A}^{aug}\boldsymbol{H}_{c'}$$

$$\boldsymbol{comp} = \boldsymbol{H}_c + \boldsymbol{H}_c'$$

Retrieved Summary-based Augmentation

Model——Attention-based Dynamic Graph Construction

计算邻接矩阵： $$\boldsymbol{A}_{v,u}^{dyn} = \frac{\mathrm{ReLU}(\boldsymbol{h}_v^T \boldsymbol{W}^Q)(\mathrm{ReLU}(\boldsymbol{h}_u^T \boldsymbol{W}^K) + \mathrm{ReLU}(\boldsymbol{e}_{v,u}^T \boldsymbol{W}^R))^T}{\sqrt{d}}$$

行归一化： $$\tilde{\boldsymbol{A}}^{dyn} = \mathrm{softmax}(\boldsymbol{A}^{dyn})$$

Model— —Hybrid Message Passing

Static Message Passing：

$$\boldsymbol{h}_v^k = \mathrm{SUM}(\{\boldsymbol{h}_u^{k-1} | \forall u \in \mathcal{N}_{(v)}\})$$

Dynamic Message Passing：

$$\boldsymbol{h}_v^{\prime k} = \sum_u \tilde{\boldsymbol{A}}_{v,u}^{dyn}(\boldsymbol{W}^V \boldsymbol{h}_u^{\prime k-1} + \boldsymbol{W}^F \boldsymbol{e}_{v,u})$$

Hybrid Message Passing：

$$\boldsymbol{f}_v^k = \mathrm{GRU}(\boldsymbol{f}_v^{k-1}, \mathrm{Fuse}(\boldsymbol{h}_v^k, \boldsymbol{h}_v^{\prime k}))$$

$$\mathrm{Fuse}(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{z} \odot \boldsymbol{a} + (1 - \boldsymbol{z}) \odot \boldsymbol{b} \quad \boldsymbol{z} = \sigma(\boldsymbol{W}_z[\boldsymbol{a}; \boldsymbol{b}; \boldsymbol{a} \odot \boldsymbol{b}; \boldsymbol{a} - \boldsymbol{b}] + \boldsymbol{b}_z)$$

Model— —Decoder

max-pooling得到的graph representation

Fuse

weighted final state of retrieved summary

Experiments

DataSet: C Code Summarization Dataset

Construction: in-domain/out-of-domain

Experiments

Table 1: Automatic evaluation results (in %) on the CCSD test set.

Methods	In-domain			Out-of-domain			Overall		
	BLEU-4	ROUGE-L	METEOR	BLEU-4	ROUGE-L	METEOR	BLEU-4	ROUGE-L	METEOR
TF-IDF	15.20	27.98	13.74	5.50	15.37	6.84	12.19	23.49	11.43
NNGen	15.97	28.14	13.82	5.74	16.33	7.18	12.76	23.93	11.58
CODE-NN	10.08	26.17	11.33	3.86	15.25	6.19	8.24	22.28	9.61
Hybrid-DRL	9.29	30.00	12.47	6.30	24.19	10.30	8.42	28.64	11.73
Transformer	12.91	28.04	13.83	5.75	18.62	9.89	10.69	24.65	12.02
Dual Model	11.49	29.20	13.24	5.25	21.31	9.14	9.61	26.40	11.87
Rencos	14.80	31.41	14.64	7.54	23.12	10.35	12.59	28.45	13.21
GCN2Seq	9.79	26.59	11.65	4.06	18.96	7.76	7.91	23.67	10.23
GAT2Seq	10.52	26.17	11.88	3.80	16.94	6.73	8.29	22.63	10.00
SeqGNN	10.51	29.84	13.14	4.94	20.80	9.50	8.87	26.34	11.93
HGNN w/o augment & static	11.75	29.59	13.86	5.57	22.14	9.41	9.98	26.94	12.05
HGNN w/o augment & dynamic	11.85	29.51	13.54	5.45	21.89	9.59	9.93	26.80	12.21
HGNN w/o augment	12.33	29.99	13.78	5.45	22.07	9.46	10.26	27.17	12.32
HGNN w/o static	15.93	33.67	15.67	7.72	24.69	10.63	13.44	30.47	13.98
HGNN w/o dynamic	15.77	33.84	15.67	7.64	24.72	10.73	13.31	30.59	14.01
HGNN	**16.72**	**34.29**	**16.25**	**7.85**	**24.74**	**11.05**	**14.01**	**30.89**	**14.50**

Experiments

Table 2: Human evaluation results on the CCSD test set.

Metrics	NNGen	Transformer	Rencos	SeqGNN	***HGNN***
Relevance	3.23	3.17	3.48	3.09	**3.69**
Similarity	3.18	3.02	3.32	3.06	**3.51**

Experiments

Table 3: Examples of generated summaries on the CCSD test set.

Example	Example 1	Example 2

Source Code	`static void strInit(Str *p){`
	`p->z = 0;`
	`p->nAlloc = 0;`
	`p->nUsed = 0;`
	`}`
		`void ReleaseCedar(CEDAR *c){`
		`if (c == NULL)`
		`return;`
		`if (Release(c->ref) == 0)`
		`CleanupCedar(c);`
		`}`

Ground-Truth	initialize a str object	release reference of the cedar
NNGen	free the string	release the virtual host
Transformer	reset a string	release of the cancel object
Rencos	append a raw string to the json string	release of the cancel object
SeqGNN	initialize the string	release cedar communication mode
HGNN	**initialize a string object**	**release reference of cedar**

Experiments

Table 4: Automatic evaluation results (in %) on the PCSD test set.

Methods	BLEU-4	ROUGE-L	METEOR
NNGen	21.60	31.61	15.96
CODE-NN	16.39	28.99	13.68
Transformer	17.06	31.16	14.37
Rencos	24.02	36.21	18.07
HGNN w/o static	24.06	38.28	18.66
HGNN w/o dynamic	24.13	38.64	18.93
HGNN	**24.42**	**39.91**	**19.48**